LEGAL NOTICE:

This erratum to the NVM Express revision 1.2 specification is proprietary to the NVM Express, Inc. (also referred to as “Company”) and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this erratum to the NVM Express revision 1.2 specification subject, however, to the Member’s continued compliance with the Company’s Intellectual Property Policy and Bylaws and the Member’s Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **“© 2007 - 2015 NVM Express, Inc. ALL RIGHTS RESERVED.”** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN “**AS IS**” BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

NVM Express Workgroup
c/o Virtual, Inc.
401 Edgewater Place, Suite 600
Wakefield, MA 01880
info@nvmexpress.org

NVM Express™ Technical Errata

Errata ID	**004**
Revision Date	**8/3/2015**
Affected Spec Ver.	**NVM Express™ 1.2**
Corrected Spec Ver.	

Errata Author(s)

Name	Company
John Carroll	Intel
Peter Onufryk	PMC
Uma Parepalli	SKHMS
Judy Brock	Samsung
Ramdas Kachare	Seagate
Suman Prakash	Samsung
Jason Gao	HGST

Errata Overview

When feature values are persistent across power states and reset is clarified.

The Invalid Format status value is clarified for Format NVM and Namespace Management.

There are several clarifications for Reservations error cases, including behaviors for Ignore Existing Key and Preempt Reservation Key.

The behavior when a NSID of FFFFFFFFh is specified for a Feature that is not namespace specific is clarified.

The behavior for end-to-end data protection for Type 1, Type 2, and Type 3 is made more explicit by separating the description for each case.

Revision History

Revision Date	Change Description
4/2/2015	First draft
4/20/2015	Edits adding error log clarifications and E2E protection
5/14/2015	Removed “globally unique” edits based on WG feedback
5/21/2015	Pushed Error log clarifications to ECN 006. Clarified Set Features behavior for features that are not namespace specific.
6/4/2015	Added more original text to 8.8.7 (for clarity), added “computed” to reference tag.
8/3/2015	Ratified.

Description of Specification Changes

Modify a portion of Figure 108 as shown below:

Figure 108: Set Features – Feature Identifiers

Feature Identifier	O/M	Persistent Across Power States and Reset[2]	Uses Memory Buffer for Attributes	Description
00h				Reserved
01h	M	No	No	Arbitration
02h	M	No	No	Power Management
03h	O	Yes	Yes	LBA Range Type
04h	M	No	No	Temperature Threshold
05h	M	No	No	Error Recovery
06h	O	No	No	Volatile Write Cache
07h	M	No	No	Number of Queues
08h	M	No	No	Interrupt Coalescing
09h	M	No	No	Interrupt Vector Configuration
0Ah	M	No	No	Write Atomicity Normal
0Bh	M	No	No	Asynchronous Event Configuration
0Ch	O	No	Yes	Autonomous Power State Transition
0Dh	O	No[3]	No[4]	Host Memory Buffer
0Eh – 77h				Reserved
78h – 7Fh		Refer to the NVMe Management Interface Specification for definition.		
80h – BFh				Command Set Specific (Reserved)
C0h – FFh				Vendor Specific[1]

NOTES:

1. The behavior of a controller in response to an inactive namespace ID to a vendor specific Feature Identifier is vendor specific.
2. This column is only valid if the feature is not saveable (refer to section 7.8). If the feature is saveable, then this column is not used and any feature may be configured to be saved across power states and reset. ~~This column is only valid if bit 4 in the Optional NVM Command Support field of the Identify Controller data structure in Figure 90 is cleared to ‘0’.~~
3. The controller does not save settings for the Host Memory Buffer feature across power states and reset events, however, host software may restore the previous values. Refer to section 8.9.
4. The feature does not use a memory buffer for Set Features, but it does use a memory buffer for Get Features. Refer to section 8.9.

O/M: O = Optional, M = Mandatory

Modify a portion of section 5.15.1 as shown below:

A completion queue entry is posted to the Admin Completion Queue when the NVM media format is complete. Format NVM command specific status values are defined in Figure 1.

Figure 1: Format NVM – Command Specific Status Values

Value	Description
Ah	**Invalid Format:** The format specified is invalid. This may be due to various conditions, including: 1) specifying an invalid LBA Format number, or 2) enabling protection information when there is not sufficient metadata per LBA, or 3) the specified format is not available in the current configuration, or 4) invalid security state (refer to TCG SIIS), etc.

Modify a portion of section 5.13.1 as shown below:

When the command is completed, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.

Namespace Management command specific status values are defined in Figure 103.

Figure 103: Namespace Management – Command Specific Status Values

Value	Description
0Ah	**Invalid Format:** The LBA Format specified is not supported. This may be due to various conditions, including: 1) specifying an invalid LBA Format number, or 2) enabling protection information when there is not sufficient metadata per LBA, or 3) the specified format is not available in the current configuration, or 4) invalid security state (refer to TCG SIIS), etc.
15h	**Namespace Insufficient Capacity:** Creating the namespace requires more free space than is currently available. The Command Specific Information field of the Error Information Log specifies the total amount of NVM capacity required to create the namespace in bytes.
16h	**Namespace Identifier Unavailable:** The number of namespaces supported has been exceeded.
1Bh	**Thin Provisioning Not Supported:** Thin provisioning is not supported by the controller.

Modify Figure 32 as shown below:

Figure 32: Status Code – Command Specific Status Values

Value	Description	Commands Affected
00h	Completion Queue Invalid	Create I/O Submission Queue
01h	Invalid Queue Identifier	Create I/O Submission Queue, Create I/O Completion Queue, Delete I/O Completion Queue, Delete I/O Submission Queue
02h	Invalid Queue Size	Create I/O Submission Queue, Create I/O Completion Queue
03h	Abort Command Limit Exceeded	Abort
04h	Reserved	Reserved
05h	Asynchronous Event Request Limit Exceeded	Asynchronous Event Request
06h	Invalid Firmware Slot	Firmware Commit
07h	Invalid Firmware Image	Firmware Commit
08h	Invalid Interrupt Vector	Create I/O Completion Queue
09h	Invalid Log Page	Get Log Page
0Ah	Invalid Format	Format NVM, Namespace Management
0Bh	Firmware Activation Requires Conventional Reset	Firmware Commit
0Ch	Invalid Queue Deletion	Delete I/O Completion Queue
0Dh	Feature Identifier Not Saveable	Set Features
0Eh	Feature Not Changeable	Set Features
0Fh	Feature Not Namespace Specific	Set Features
10h	Firmware Activation Requires NVM Subsystem Reset	Firmware Commit
11h	Firmware Activation Requires Reset	Firmware Commit
12h	Firmware Activation Requires Maximum Time Violation	Firmware Commit
13h	Firmware Activation Prohibited	Firmware Commit
14h	Overlapping Range	Firmware Commit, Firmware Image Download, Set Features
15h	Namespace Insufficient Capacity	Namespace Management
16h	Namespace Identifier Unavailable	Namespace Management
17h	Reserved	
18h	Namespace Already Attached	Namespace Attachment
19h	Namespace Is Private	Namespace Attachment
1Ah	Namespace Not Attached	Namespace Attachment
1Bh	Thin Provisioning Not Supported	Namespace Management
1Ch	Controller List Invalid	Namespace Attachment
1Dh – 7Fh	Reserved	
80h – BFh	I/O Command Set Specific	
C0 – FFh	Vendor Specific	

Modify Figure 179 as shown below:

Figure 2: Reservation Acquire – Command Dword 10

Bit	Description
31:16	Reserved
15:08	**Reservation Type (RTYPE):** This field specifies the type of reservation to be created. The field is defined in **Error! Reference source not found.**.
07:04	Reserved
03	**Ignore Existing Key (IEKEY)**: If this bit is set to a '1' and the host is a registrant, then the Current Reservation Key (CRKEY) check is disabled and the command shall succeed regardless of the CRKEY field value. If this bit is set to '1' and the host is not a registrant, the controller should return an error of Invalid Field In Command. If this bit is cleared to '0', then the Current Reservation Key is checked.

Modify Figure 186 as shown below:

Figure 3: Reservation Release – Command Dword 10

Bit	Description
31:16	Reserved
15:08	**Reservation Type (RTYPE):** If the Reservation Release Action is 00b (i.e., Release), then this field specifies the type of reservation that is being released. The reservation type in this field shall match the current reservation type; if it does not match the controller should return an error of ~~Reservation Conflict~~ Invalid Field In Command. This field is defined in **Error! Reference source not found.**.

Modify a portion of section 8.8.7 as shown below:

8.8.7 Preempting a Reservation or Registration

A host that is a registrant may preempt a reservation and/or registration by executing a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 001b (Preempt), and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field. A host that is a registrant may preempt without regard to its current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the Reservation Register command. The preempt actions that occur are dependent on the type of reservation held on the namespace, if any, and the value of the Preempt Reservation Key (PRKEY) field in the command. If the host is not a registrant, then the command is aborted with a status of Reservation Conflict. The remainder of this section assumes that the host is a registrant.

If the existing reservation type is not Write Exclusive - All Registrants and not Exclusive Access - All Registrants, then the actions performed by the command depend on the value of the PRKEY field as follows. If the PRKEY field value matches the reservation key of the current reservation holder, then the following occur as an atomic operation: the reservation holder is unregistered, the reservation is released, and a new reservation is created of the type specified by the Reservation Type (RTYPE) field in the command for the host as the reservation key holder. If the PRKEY field value does not match that of the current reservation holder and is not equal to zero, then registrants whose reservation key matches the value of the PRKEY field are unregistered. If the PRKEY field value does not match that of the current reservation holder and is equal to zero, then the command is aborted with status Invalid Field in Command.

If the existing reservation type is Write Exclusive - All Registrants or Exclusive Access - All Registrants, then the actions performed by the command depend on the value of the PRKEY field as follows. If the PRKEY field value is zero, then the following occurs as an atomic operation: all registrants other than the host that issued the command are unregistered, the reservation is released, and a new reservation is created for the host of the type specified by the Reservation Type (RTYPE) field in the command. If the PRKEY value is non-zero, then registrants whose reservation key matches the value of the PRKEY field are unregistered. If the PRKEY value is non-zero and there are no registrants whose reservation key matches the value of the CRKEY field, the controller should return an error of Reservation Conflict.

If there is no reservation held on the namespace, then execution of the command causes registrants whose reservation key match the value of the PRKEY field to be unregistered.

A reservation holder may preempt itself using the above mechanism. When a host preempts itself the following occurs as an atomic operation: registration of the host is maintained, the reservation is released, and a new reservation is created for the host of the type specified by the RTYPE field.

A host may abort commands as a side effect of preempting a reservation by executing a Reservation Acquire command and setting the RACQA field to 010b (Preempt and Abort). The behavior of such a command is exactly the same as that described above with the RACQA field set to 001b (Preempt), except that commands that target the namespace are aborted by controllers associated with hosts whose reservation or registration is preempted. As with the Abort Admin command, abort as a side effect of preempting a reservation is best effort; the commands to abort may have already completed, currently be in execution, or may be deeply queued.

Modify a portion of section 7.8 as shown below:

Feature settings may apply to the entire controller (and all associated namespaces) or may apply to each namespace individually. To change or retrieve a value that applies to the controller and all associated namespaces, host software sets CDW1.NSID to 0h or FFFFFFFFh in the Set Features or Get Features command. Features that are not namespace specific shall have the CDW1.NSID field set to 0h.

<insert blank line>

To change or retrieve a value that applies to a specific namespace, host software sets CDW1.NSID to the identifier of that namespace in the Set Features or Get Features command. If host software specifies a valid CDW1.NSID value that is not 0h or FFFFFFFFh and the Feature is not namespace specific, then a Set Features command returns the Feature Not Namespace Specific status code, whereas a Get Features command ~~the controller~~ returns the Feature value that applies to the entire controller.

Modify a portion of section 8.3 as shown below:

Checking of protection information consists of the following operations performed by the controller. If bit 2 of the Protection Information Check (PRCHK) field of the command is set to ‘1’, then the controller compares the protection information Guard field to the CRC-16 computed over the logical block data. If bit 1 of the PRCHK field is set to ‘1’, then the controller compares unmasked bits in the protection information Application Tag field to the Logical Block Application Tag (LBAT) field in the command. A bit in the protection information Application Tag field is masked if the corresponding bit is cleared to ‘0’ in the Logical Block Application Tag Mask (LBATM) field of the command.

For Type 1 protection, i~~I~~f bit 0 of the PRCHK field is set to ‘1’, then the controller compares the protection information Reference Tag field to the computed reference tag. The value of the computed reference tag for the first LBA of the command is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command~~, for writes and reads respectively~~. The computed reference tag is incremented for each subsequent logical block. Unlike SCSI Protection Information Type 1 protection which implicitly uses the least significant four bytes of the LBA, ~~T~~the controller always uses the ILBRT or EILBRT field and requires host software to initialize the ILBRT or EILBRT field to the least significant four bytes of the LBA when Type 1 protection is used. If the ILBRT or EILBRT field does not match the least significant four bytes of the LBA, then the controller completes the command with an Invalid Protection Information status code.

For Type 2 protection, if bit 0 of the PRCHK field is set to ‘1’, then the controller compares the protection information Reference Tag field from each logical block to the computed reference tag. The computed reference tag is incremented for each subsequent logical block. The value of the computed reference tag for the first LBA of the command is the value contained in the ILBRT or EILBRT field in the command. Host software may set the ILBRT and EILBRT fields to any value.

For Type 3 protection, if bit 0 of the PRCHK field is set to ‘1’, then the command may be aborted with status Invalid Field in Command. The controller may ignore the ILBRT and EILBRT fields when Type 3 protection is used because the computed reference tag remains unchanged.